

Genetic Algorithm For Restrictive Channel Routing Problem

Vladimir N. Davidenko, Victor M. Kureichik, Victor V. Miagkikh[†]

Taganrog State University of Radio Engineering,
Nekrasovsky 44, GSP-17A,
Taganrog, 347928, Russia
Phone: (+7-86344) 6-1768
email: kur@tsure.ru

Michigan State University[†]
217 Engineering Building
East Lansing, MI, 48823
Phone: (517) - 355-4553
e-mail: miagkikh@cps.msu.edu

Abstract

This paper presents a genetic algorithm for the restrictive channel routing problem. The major difference of the algorithm from already known genetic algorithms for this problem consists of the use of information about horizontal and vertical constraints in chromosome encoding that allow prevention of “illegal” solutions. This representation leads to lower complexity, because solution repairing procedures become unnecessary. Moreover, the search space is greatly reduced. Competitive experimental results proving the consistency of the approach were obtained.

Genetic Algorithm For Restrictive Channel Routing Problem

Abstract

This paper presents a genetic algorithm for the restrictive channel routing problem. The major difference of the algorithm from already known genetic algorithms for this problem consists of the use of information about horizontal and vertical constraints in chromosome encoding that allow prevention of "illegal" solutions. This representation leads to lower complexity, because solution repairing procedures become unnecessary. Moreover, the search space is greatly reduced. Competitive experimental results proving the consistency of the approach were obtained.

1. INTRODUCTION

Routing is the final stage in physical design of Integrated Circuits (IC). The aim of routing is to find a topology of connections among elements of an IC in accordance with a given net list, along with other design requirements and preferences. Routing is one of the most highly constrained problems in IC design. There exist a large number of various methods for its solution, such as wave and beam routing algorithms and their various modifications. These algorithms require a large amount of computing resources and are frequently unable to find near-optimal solutions for complex problems. Moreover, the obtained solutions often require re-routing.

Since the solution of the general routing problem is very difficult, the whole routing area should be decomposed into a number of rectangular areas called channels. After its introduction in 1972 by Hashimoto and Stevens [12] this problem has been intensively investigated. Near-optimality, reliability and efficiency are the main reasons for the popularity of channel routers today. They require smaller computational resources and are capable of finding near-optimal solutions. In spite of complexity reduction in comparison to the general routing problem, channel routing remains NP-complete [14]. There exist various approaches to channel routing using various conditions of routing [1-3]. In this paper, we present a genetic algorithm for the Restrictive Channel Routing Problem (RCRP).

Genetic Algorithms (GA) is a global evolutionary search technique known to be efficient in the solution of NP-

complete problems [7,9,10]. They were successfully used in VLSI design problems like placement [4,5,8] and channel routing [6,11,13]. The major difference of the proposed algorithm lies in the use of information about horizontal and vertical constraints in chromosome encoding, which allows it to exclude occurrences of "illegal" solutions violating one or more constraints, and improve search quality.

There exist several applications to routing problems based on GA and other evolutionary computation techniques [6,11,13]. Lio *et al* [6] used evolutionary programming to approach the RCRP. Rahmani and Ono [11] developed a specialised mutation operator for RCRP based on vertical and horizontal constraint graphs. Lattice-like chromosome encoding and specialized genetic operators were used by Lienig and Thulasiraman [13] to solve the channel routing problem. However, this algorithm uses doglegs not allowed in RCRP, that complicates comparison.

A brief introduction to the restrictive channel routing problem will be presented in Section 2. Section 3 discusses the application of the GA to the problem and describes the use of the heuristic information based on the graphs of vertical and horizontal constraints. Experimental results supporting the validity of the proposed approach will be provided in Section 4 followed by conclusions in Section 5.

2. RESTRICTIVE CHANNEL ROUTING PROBLEM

In the classical channel routing problem the routing area (*channel*) is modelled on a rectangular grid with terminals on the top and bottom sides only (see Fig. 1). The distance between the two channel boundaries called *channel width* is determined by the intensity of the wiring in the channel. *Track* is a segment of line along which the connection in the primary direction can go or, in other words, a row in the grid modeling the actual channel.

Each *net*, i.e. the connection among corresponding contacts, is presented as a single horizontal segment with several vertical ones [2]. Horizontal segments are placed in one layer, and vertical ones in another. The connections between these horizontal and vertical segments are made through *vias*. The classical statement of the channel routing problem consists of finding the least-width channel that is sufficient for the placement of all connections. In addition, the total

length of nets, number of vias, etc. are the parameters being minimized.

The RCRP is a simple model for a rectangular channel with two terminals of contacts along its top and bottom sides. The *doglegs*, i.e. the transition of a horizontal segment from one track to another, are not allowed.

Let us proceed to the formal statement of the problem. A channel is described by two sequences, *Top* and *Bottom*, in which the top and the bottom rows of the channel are placed respectively [6]. The size of both sequences is C (the number of columns in the channel). The set of nets is defined as $Net = \{N_1, \dots, N_n\}$, where n is the number of nets. E.g. $Top = \{1, 0, 3, 1, 4, 2, 3, 2\}$; $Bottom = \{6, 4, 6, 6, 3, 0, 5, 5\}$, where $C=8$, $n=6$, $Net = \{1, 2, 3, 4, 5, 6\}$, the element 0 in *Top* or in *Bottom* denotes empty contact. The layout of the channel with connected nets is shown in Fig. 1.

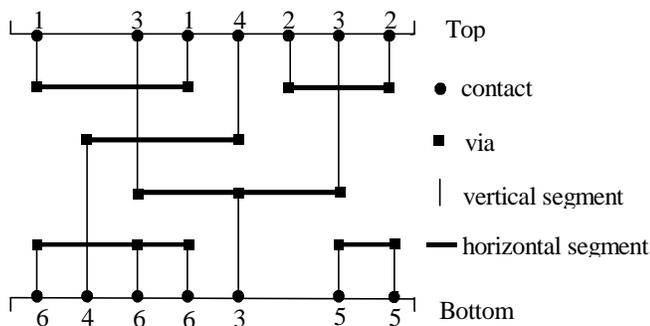


Fig. 1

In order to prevent overlaps of vertical and horizontal segments, graphs of vertical and horizontal constraints can be used [1]. Vertical constraints are described by the directed Vertical Constraints Graph $G_V = (E_{Net}, E_V)$, where E_{Net} is the set of vertices, corresponding to the set of the nets and E_V is the set of links. Edge (n, m) in E exists if and only if net n is located above net m to prevent crossings of the vertical segments of nets [15]. E.g. in G_V in Fig. 2, there is a path from node 1 to node 6 which means that net 1 must be placed above net 6 in order to avoid crossing of vertical segments for contacts 1 and 4 (see Fig. 1). G_V should be *acyclic* to solve the problem within the framework of the RCRP [15]. Otherwise, the problem can be solved with the introduction of *doglegs*, which are not permitted in the RCRP.

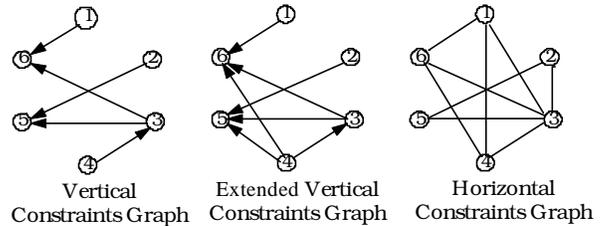


Fig. 2

In this paper, we will use an Extended Vertical Constraints Graph $G_{EV} = (E_{Net}, E_{EV})$, where E_{Net} is the set of vertices corresponding to the set of nets and E_{EV} is the set of links. Edge (n, m) in E_{EV} exists if and only if there is a path from vertex n to vertex m in G_V . For example, in Fig. 2, there is a path from node 4 to node 5 through node 3. Accordingly, there exists an edge $(4,5)$ in G_{EV} . Horizontal constraints are presented by the undirected Horizontal Constraints Graph $G_H = (E_{Net}, E_H)$, where E_{Net} is the set of nets and E_H is the set of edges. Edge (n, m) in E_H exists if and only if n and m must be routed in different tracks to avoid the overlap of horizontal segments n and m . For example, there exists an edge $(1,4)$ meaning that net 1 cannot be placed in the same track with net 4 (see. Fig. 1).

3. A GENETIC ALGORITHM FOR RESTRICTIVE CHANNEL ROUTING

Genetic Algorithm (GA) is an iterative procedure, which processes a group of individuals (solutions) called a *population* [7]. Each individual has a *chromosome* encoding a possible solution. GA, as any other search technique, requires some criterion for the evaluation of the solutions. This criterion, called the *fitness function*, estimates the quality of the individual. Various genetic operators, such as *crossover* and *mutation*, with some probabilities P_C and P_M respectively, are consequently applied to solutions picked from the population by means of the *selection* operator. The best parents and the offspring make up the population in subsequent iterations of the algorithm, called *generations*.

This paper uses a novel topological description of the channel. Thus, a chromosome describes the relative positions of the nets. For this purpose each pair of nets (m, n) corresponds to a gene, which can have three possible values:

- 0 - net m should be placed above net n ,
- 1 - net m should be placed below net n
- * - relative position is irrelevant to relative positions of the other nets.

The last instance occurs if the nets have no horizontal constraints. In the example depicted in Fig. 1, the chromosome will have the following form:

Net m	1	1	1	1	1	2	2	2	2	3	3	3	4	4	5
Net n	2	3	4	5	6	3	4	5	6	4	5	6	5	6	6
Gene	*	0	0	*	0	0	*	0	*	1	0	0	*	0	*

The length of the chromosome in such encoding is rather large and equals $L = \sum_{i=0}^{N-1} i$. The analysis of G_{EV} and G_H can lead to a reduction of this length. The decrease is possible because relative positions of some pairs in nets are already fixed in G_{EV} . A change of these positions results in constraint violations. The second observation allowing reduction of the length of a chromosome is the following. If nets have no horizontal constraints, then their relative positions are not important, or can be figured out from the relations among other nets. This fact is denoted as a "*" sign in the representation above. Thus, the length of a chromosome is $L = N_{GC} - N_{EVC}$, where N_{GC} is the number of horizontal constraints in G_H and N_{EVC} is the number of vertical constraints imposed by G_{EV} . For our example, pairs of nets (1,6), (2,5), (3,4), (3,5), (3,6), (4,6) have relative positions determined by G_{EV} ; and relative positions of the nets (1,2), (1,5), (2,6), (4,5), (5,6) are irrelevant, since the nets in these pairs do not have horizontal constraints. Therefore, the length of the chromosome will be equal to 3 and look like this:

Net m	1	1	2
Net n	3	4	3
Gene	0	0	0

To receive an instance of a channel from a chromosome, we use directed Topology Graph $G_T = (E_{Net}, E_T)$, where E_{Net} is the set of nets; E_T is the set of links describing relative positions of the nets in the channel. Edge (m, n) in E_T exists if and only if net m is located in the channel above net n , i.e. on a track with smaller number. G_T is retrieved from the chromosome in accordance with the following algorithm in pseudocode:

- Step 1.** Copy G_{EV} to G_T .
- Step 2.** $i = 1$.
- Step 3.** If edge (m_i, n_i) does not belong to G_T , than add it to G_T .
- Step 4.** For all nodes k ($k = 1, \dots, N$), such that $k \neq m_i$ and $k \neq n_i$, and edges (m_i, k) not existing in G_T , check the existence of a path from node m_i to node k . If such path exists than add edge (m_i, k) to G_T .
- Step 5.** If $i \leq L$, than increment i and go to Step 3.
- Step 6.** Formation of G_T completed.

The following algorithm restores the channel from the chromosome:

- Step 1.** Retrieve G_T from the chromosome.
- Step 2.** $i = 0$.
- Step 3.** Find nodes which have only outgoing links, and place them on track with number i and delete these nodes from the graph G_T .
- Step 4.** Increment i .

- Step 5.** If G_T is not empty, then go to Step 3.
- Step 6.** Without breaking relative positions, place the nets on tracks in a way that has minimal length of the vertical segments of nets.
- Step 7.** Return obtained draft of the placement and channel of width i .

For example, let $A = 000$. Fig. 1 shows obtained routing, which is the optimal solution for this example. If $A=010$, the layout of the channel with connected nets for such a chromosome is shown in Fig. 3.

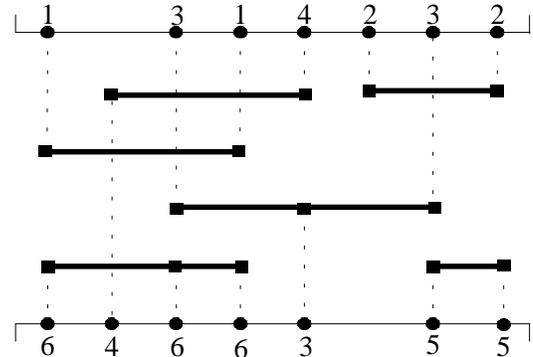


Fig. 3

A simple fitness function applicable to this problem is:

$$F(A) = 2(UsedTrack(A) + 2)C + TotalVertSeg(A),$$

where C is the number of contacts and function $UsedTrack(A)$ returns the number of tracks occupied by the channel in the received solution. The function $TotalVertSeg(A)$ calculates the length of the vertical segments of nets in the obtained solution. The length of the vertical segments of net is determined by the distance between contacts and vias, which connects vertical and horizontal segments. For instance, the channel in Fig. 1 has the number of used tracks equal to 4; the length of vertical segments equals 22. Thus, the fitness of the chromosome equals $F(A) = (4 + 2) * 8 + 22 = 70$. The fitness of the channel shown in Fig. 3 is 72. This fitness function is primarily directed to the minimization of width of the channel and the total length of connections. There also exist different evaluation functions for this problem [6,11]. They were used respectively in comparison of the results with these algorithms. The evaluation function above is more computationally simple in comparison to the functions used there, but takes into account the same factors.

After selection, all chromosomes are grouped into pairs, and then the crossover operation with probability P_C is applied to each pair. After recombination, offspring are exposed to mutation with probability P_M . For our chromosome encoding, the classical one- and two-point crossover operators are well suited. Having considered one-point and two-point

crossovers, the latter was chosen. The following example illustrates the recombination process:

Parent 1: 0 | 1 | 0 fitness = 72
 Parent 2: 1 | 0 | 1 fitness = 77
 Offspring 1: 0 | 0 | 0 fitness= 70
 Offspring 2: 1 | 1 | 1 fitness= 77

In this example, the offspring, namely Offspring 1, has better fitness than fitness of the parents. Parent 1 is shown in Fig. 3, Parent 2 - in Fig. 4, Offspring 1 is shown in Fig. 1, and Offspring 2, after decoding of the chromosome, looks the same as Parent 2.

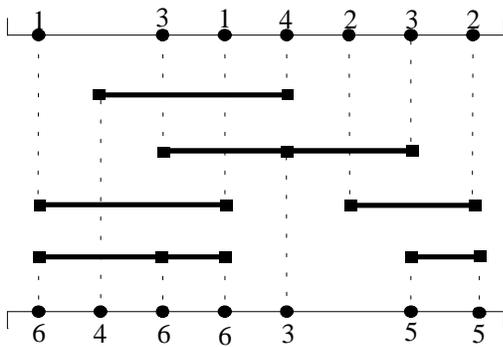


Fig. 4

The mutation used in this work inverts an arbitrary gene in a chromosome with probability P_M . This example shows this mechanism:

Before mutation: 0 (1) 0 fitness = 72
 After mutation: 0 (0) 0 fitness = 70,

where (x) is mutation point. The channel before the application of the operation of mutation is shown in Fig. 3. The channel after application of operation of mutation is shown in Fig. 1.

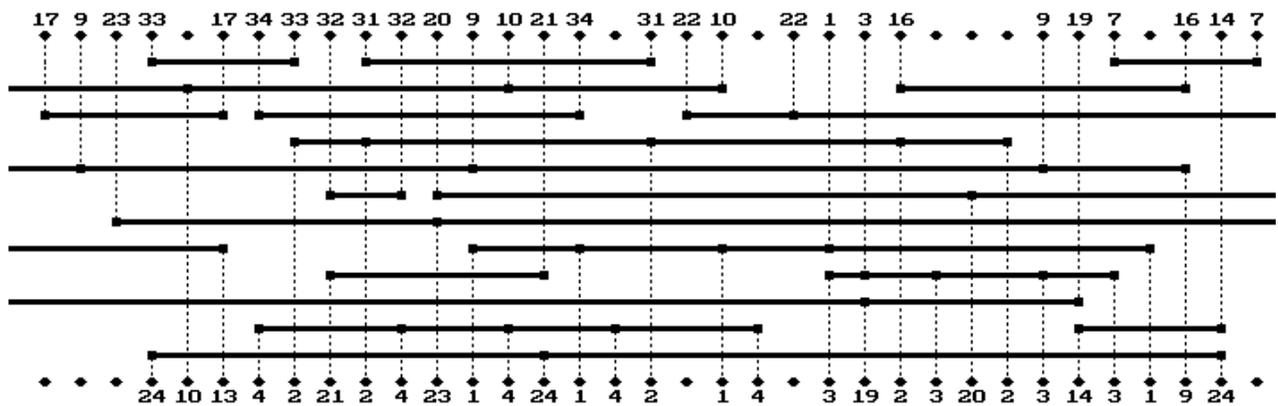


Fig. 5.

4. EXPERIMENTAL RESULTS

To illustrate the validity of our approach described in the previous sections, a C++ program was developed to compare the performance of our algorithm with other algorithms. Authors have chosen three benchmark problems taken from [1] (tests Ex1, Ex3b, Ex3c). The minimal numbers of tracks found by the different algorithms are given in Table 1, where EA is Efficient Algorithms for Channel Routing [1], EP is Evolution Programs for Restrictive Channel Routing [6], GA1 - Genetic Algorithm for Channel Routing Problem [11], GA2 is presented algorithm.

Table 1.

Benchmark	EA	EP	GA1	GA2
Ex1	12	12	12	12
Ex3b	17	17	17	17
Ex3c	18	18	18	18

The optimal solutions or solutions with a deviation from the optimal solution on one track were found during the first 20 generations in all cases. The optimal solution was produced during 100-200 generations. E.g., the best solution for Ex1 benchmark is shown in Fig. 5. The search capabilities of our algorithms are comparable with the results of other known methods, however runtime is much faster. Mentioned algorithms have to use special procedures for resolving the vertical and horizontal violations, which are not necessary using our encoding. In our case, the complexity of the restoration of the channel layout from a chromosome is $O(N^2)$. Thus, the time complexity of the whole algorithm is $T * (M * P_C * 2) * P_M * O(N^2)$, where M is the size of a population, T is the number of generations. Thus, the complexity of the whole algorithm is $O(N^2)$.

In this experiment, the probability of crossover P_C was equal to 1. The size of population M was equal to 50. A rather high probability of mutation $P_M=0.1$ has been chosen in accordance with our experiments on finding the optimal

probabilities described below. “Elite” selection was used. Then, on average, after five generations all possible solutions having the minimal number of tracks were obtained. Solutions with less than 1 % deviation from the optimal result taking into account the length of connections were evolved during twelve generations. Ten runs were used to obtain these statistics. Unfortunately, a direct comparison of the run times was not possible due to difference of hardware used in runs.

Table 2 shows the optimal results for the chosen examples and the average deviations from the optimal results. The averages were obtained by averaging the results of 10 runs using various random seeds. The tests were carried out with $M=50$, $T=20$, $P_C=1.0$ and $P_M=0.1$.

Then we tried to find optimal values for the probabilities of crossover and mutation which minimize the average deviation from optimal results. Table 2 shows that the most

Table 2

Number of test	Number of terms	Number of nets	Number of tracks	Optimal length of nets	Average deviation from optimal result, %
0	80	51	8	773	0,0006
1	80	38	10	1072	0,0000
2	80	36	9	1076	0,0005
3	80	37	10	1134	0,0000
4	100	66	10	1282	0,9204

difficult example was Test 4. Therefore, authors have tried to find the values of the probabilities of crossover and mutation for it. For this purpose the value of probability of mutation $P_M=0.10$ was set, and the value of crossover probability was changing from 0 up to 1 with a step 0.10. 10 tests for each value of P_C were performed. The results are summarized in Fig 6. It shows that the minimal deviation was found when probability of crossover was equal to 1.

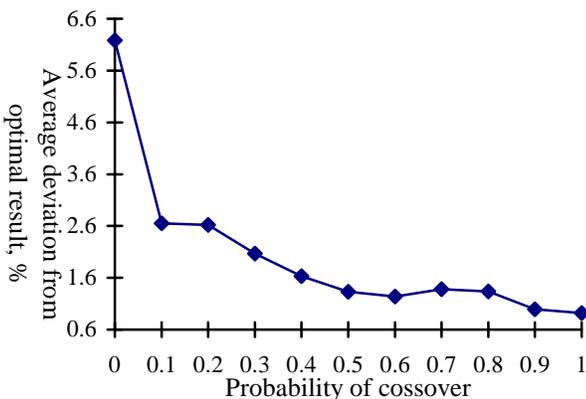


Fig. 6.

Then we tried to find the optimal value for the probability of mutation. For this purpose authors set the value $P_C=1$ and changed the value for the probability of mutation from 0 to 1 with step 0.10, and again performed 10 runs for each

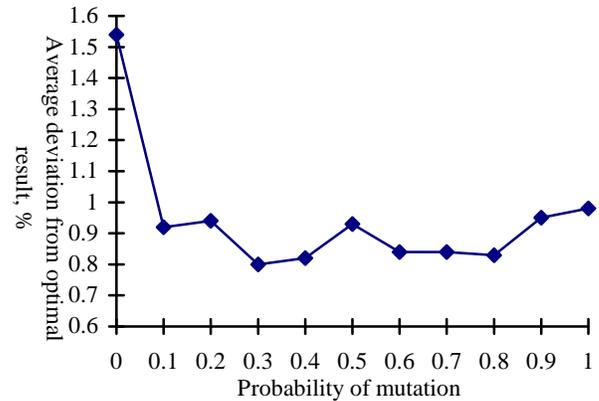


Fig. 7.

value. Results are presented in Fig. 7. Tests have shown that the best results were found with the value $P_M=0.30$. Average deviations from the optimal solution was 0.8 %.

Experimental results on various problems demonstrated the robustness of the algorithm. Faster run time and low complexity are accounted for by the new chromosome encoding. This representation made solution repairing procedures unnecessary.

5. CONCLUSION

In this paper, the application of genetic algorithm to the restrictive channel routing problem was investigated. The results of experiments have shown that the presented algorithm obtains near-optimal solutions in a short period of time. These benefits are attributed to the proposed chromosome encoding, which is derived from a mathematical analysis of the problem. As a result, the search space is reduced greatly by excluding those solutions which cannot be accepted in advance due to constraint violations. This fact is the main advantage of the presented algorithm resulting in its better performance. These findings are another illustration of the fact that incorporation of domain-specific knowledge into appropriate encoding schemes and genetic operators is very powerful source of GA performance enhancement [4].

Though the current results are good, further improvements are possible through application of gene sorting and the use of customized genetic operators and other problem-specific information. These points are under investigation now.

REFERNCES

- [1] Yoshimura, T. and Kuh, E.S. "Efficient algorithms for channel routing", *IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst.*, vol.1, no.1, pp.25-35,1982.
- [2] Burstein, M., "Channel routing", *Layout Design and Verification*, pp. 133-167, Elsevier Science, 1986.
- [3] Shimamoto, T. and Sakamoto, A., "Neural computation for channel routing using Hopfield neural network model", *Trans. IEICE*, vol.E72, no.12, pp.1360-1366, 1989.
- [4] Michalewicz, Z., "Genetic Algorithms + Data Structures = Evolution Programs", *Springer-Verlag*, 1992.
- [5] Cohoon, J.P. and Paris, W.D., "Genetic placement", *IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst.*, vol.6, no.6, pp.956-964,1987.
- [6] Liu, X., Sakamoto, A., Shimamoto, T., "Restrictive Channel Routing with Evolution Programs", *Trans. IEICE*, vol.E76-A, no.10, pp.1738-1745, 1993.
- [7] Goldberg, D.E. "Genetic Algorithm in Search, Optimisation & Machine Learning", *Addison-Westley*, 1989.
- [8] Goodman, E.D., Tetelbaum, A.Y. and Kureichik, V.M., "A Genetic Algorithm Approach to Compaction, Bin Packing, and Nesting Problems", *Case Center Technical Report #940702*, Michigan State University, 1994.
- [9] Punch, W.F. Goodman, E.D., et al, "Further Research on Feature Selection and Classification Using Genetic Algorithms", *Proc. of the 5th Int. Conf. on Genetic Algorithms*, July 1993, pp. 557-564.
- [10] Goodman, E.D. and Sannier, A.P., "Genetic Learning Procedures in Distributed Environments", *Proc. Second Int'l Conf. on Genetic Algorithms and their Applications Laurence Erlbaum Associates, Inc.*, Cambridge, MA, 1987.
- [11] Rahmani, A.T. and Ono N., "A Genetic Algorithm for Channel Routing Problem", in *Proc. 5th Intl. Conf. on GAs*, pp. 494-498, 1993.
- [12] Hashimoto A., Stevens S. "Wire routing by optimising channel assignment within large apertures", In *8th Design Automation Conference*, pages 155-169, ACM/IEEE, 1971.
- [13] Lienig J, Thulasiramn K, "A genetic Algorithm for Channel Routing in VLSI Circuits", *Evolutionary Computation* 1(4), 1994
- [14] LaPaugh A.S. Algorithms for integrated Circuit Layout: an Analytic Approach. *PhD thesis*, Laboratory for Computer Science, MIT, Cambridge, MA, November 1980.
- [15] Groeneveld P., "Necessary and Sufficient Conditions for the Routability of Classical Channels", *Integration*, the VLSI Journal 16 (193), pp. 59-74.